



Towards Energy-Efficient Storage Servers

Vinícius Machado, Amanda Braga, Natália Rampon, Jean Bez, Francieli Boito, Rodrigo Kassick, Edson Padoin, Julien Diaz, Jean-François Méhaut, Philippe Navaux

► To cite this version:

Vinícius Machado, Amanda Braga, Natália Rampon, Jean Bez, Francieli Boito, et al.. Towards Energy-Efficient Storage Servers. 32nd ACM Symposium on Applied Computing (SAC 2017), Sung Y. Shin, Apr 2017, Marrakesh, Morocco. hal-01519168

HAL Id: hal-01519168

<https://hal.science/hal-01519168>

Submitted on 5 May 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Towards Energy-Efficient Storage Servers

Vinícius Machado¹, Amanda Braga¹, Natália Rampon¹, Jean Bez¹, Francieli Boito², Rodrigo Kassick¹, Edson Padoin³, Julien Diaz⁴, Jean-François Méhaut⁵, Philippe Navaux¹

¹Institute of Informatics – Federal University of Rio Grande do Sul – Porto Alegre, Brazil
{vrmachado,abbraga,ngrampon,jlbez,rkassick,navaux}@inf.ufrgs.br

²Department of Informatics and Statistics - Federal University of Santa Catarina – Florianópolis, Brazil
francieli.boito@posgrad.ufsc.br

³Regional University of the Northwest of Rio Grande do Sul (UNIJUI) – Ijuí, Brazil
padoin@unijui.edu.br

⁴Inria Project-Team Magique 3D – Inria Bordeaux Sud-Ouest – France
julien.diaz@inria.fr

⁵CNRS/Grenoble Informatics Laboratory – University of Grenoble Alpes – Grenoble, France
jean-francois.mehaut@imag.fr

ABSTRACT

As large-scale parallel platforms are deployed to comply with the increasing performance requirements of scientific applications, a new concern is getting the attention of the HPC community: the power consumption. In this paper, we aim at evaluating the viability of using low-power architectures as file systems servers in HPC environments, since processing power is of less importance for these servers. We present a performance and energy efficiency study of such low-power servers when compared to conventional architectures. Our results indicate that the low-power alternative could be a viable choice to save energy by up to 85% while not compromising on performance, specially for read-intensive workloads. We show the low-power server provides 7 times more energy efficiency to the system while running a real application from the seismic wave propagation field.

Keywords

high performance computing, low-power processors, parallel file system, energy efficiency, ARM

1. INTRODUCTION

Scientific applications – such as climate and seismic simulations – feed the high-performance computing (HPC) field with performance requirements to provide an understanding of complex phenomena. These performance requirements justify the appearance of ever increasing large scale parallel platforms. This processing power comes at the cost of high power demands. A 20MW limit on power demand was sug-

gested for future exascale systems [14], but projections based on current technologies point that, in order to meet this budget, processors’ energy efficiency needs to be increased by two orders of magnitude [17]. Therefore, power consumption has become a primary concern of the HPC community.

Seeking for alternatives to improve energy efficiency, some researchers have been studying the use of low-power architectures, where regular, performance-focused, processors are replaced by Advanced RISC Machines (ARM) processors [22]. Despite providing lower performance, these architectures provide better efficiency to some classes of scientific applications [20].

The processing units are responsible for most of the power demand in typical computational systems. Nevertheless, the gap between processing and data access speeds causes many applications to spend most of their time performing input/output (I/O) operations [4, 19]. For this reason, increasing the energy efficiency of the I/O subsystem is also an important step to tackle the energy and power challenge.

In the HPC scenario, I/O operations are typically made to the parallel file system (PFS), which is deployed over a set of dedicated machines operating as data servers. These servers receive requests from processing nodes and process them by accessing storage devices. Therefore, the processing capability of these machines is not the most relevant aspect due to the elevated time spent with I/O operations. Hence, it could be possible to employ low-power architectures as data servers to achieve better energy efficiency.

In this paper, we study the viability of replacing regular servers by low-power ones to act as parallel file system data servers. Some initiatives using such low-power architectures in data centers and cloud infrastructures have been gaining attention [23, 26]. However, we could not find any work from the literature where this alternative is evaluated and its impact on performance is directly measured.

Therefore our main contribution is evaluating both energy consumption and performance of a low-power PFS data server under representative access pat-

terns. We compare the low-power server with a regular one in order to point the impact of adopting the low-power architecture. In addition to synthetic benchmarks, **we also validate our conclusions using a real application from the seismic wave propagation domain.**

The rest of this paper is organized as follows. The next section discusses related work. Section 3 describes the methodology for this study, and results are presented in Section 4. Section 5 concludes the paper and points future work.

2. RELATED WORK

Many applications spend most of their execution time in I/O operations, hence considerable research work has focused on these operations’ energy consumption. Alan et al. [1] propose new algorithms for data transfers aiming at saving energy at end systems (sender and receiver nodes), as these are responsible for at least one-quarter of the data transfer energy consumption. Ge [8] evaluates performance and energy efficiency of parallel I/O operations with different file system configurations and applications access patterns.

In the past, researchers have explored the use of multiple speeds hard disks for storage servers [3, 10, 29]. More recent approaches apply mainly dynamic voltage and frequency scaling (DVFS) to decrease energy consumption by the processor during I/O operations, as they do not require a lot of processing power. DVFS has been a popular technique to save energy consumption by idle cores on multi-core systems [15], idle resources in cloud environments [11, 28], and during MPI applications’ communication phases [21].

Ge et al. [9] propose a technique for HPC architectures which performs DVFS in the processing nodes during their I/O operations to the parallel file system. Their technique considers characteristics of the applications to decide the best frequency. A similar approach had previously been applied to sequential applications by Shang and Wang [25].

Saito et al. [24] perform DVFS to processing nodes during checkpointing operations to local storage devices. Ibrahim et al. [12] study performance and energy consumption of MapReduce applications in cloud environments using different DVFS strategies. Chandrasekar et al. [4] propose a checkpointing/restart framework which focuses on providing both energy savings and performance improvements.

Amur et al. [2] present a distributed file system where primary replicas of data are stored in servers which are always available, whereas other replicas are placed on servers which can be turned off or on according to demand. Kim et al. [13] develop and evaluate algorithms to define the best subset of servers to be kept on or off in order to ensure data availability while minimizing energy consumption.

Nijim et al. [18] combine flash-based storage devices (SSDs) with hard disks (HDDs) to provide storage with lower energy consumption. This is achieved by using the SSDs as a cache for the HDDs. This hybrid storage strategy is explored by several works to provide high performance for I/O servers [27]. These works use the SSD only as a cache because of its higher cost per byte, which has been hampering the full replacement of hard disks in large scale architectures.

Differently from previous work discussed in this section, in this paper, we evaluate the use of low-power, ARM-based architectures as data servers for parallel file systems. Although there have been initiatives to use these low-power architectures in data centers and cloud infrastructures [23, 26], we could not find any work where this option is evaluated and has its impact on performance directly measured.

3. METHODOLOGY

This section describes the methodology adopted to achieve the objectives of this research. The selected low-power machines are two identical CubieTrucks [5] – each powered with an A20 dual-core ARM Cortex-A7 processor produced by AllWinner running at 1GHz frequency. Each board has 2GB of RAM memory and 8GB of NAND storage. We have decided to use SSDs for storage, considering that these devices are expected to replace HDDs in future systems due to their advantages such as lower data access latencies. Moreover, using a faster storage device increases the impact on performance of other components of the architecture. The used SSDs are two identical 840 Series MZ-7TD500BW by Samsung, each with a capacity of 500GB.

This low-power architecture was compared to a regular one, in this case, a computer powered with an Intel i5-4460 processor running at 3.2GHz with 8GB of RAM memory. This machine referred to as “PC”, was chosen to prioritize the conduction of experiments with a methodology as close as possible to the one used in the CubieTrucks, i.e., using the same SSD and the same equipment for power measurements. Hence, a more fair comparison is made, as we rule out eventual particularities of the used storage device and possible measurement problems.

All machines run Linux, kernel 3.7.61+ in the CubieTrucks and 4.4.0-38 in the PC. The SSDs were connected through a SATA II interface in the CubieTrucks and through a SATA III interface in the PC. Having a faster interface in the PC does not compromise the comparison, as it only gives an advantage to our baseline and thus makes the analysis more pessimistic. On both architectures, the disks were mounted with the *ext4* local file system. The operating system is located on an HDD device in the PC and on the NAND storage in the CubieTrucks, to minimize interference from the OS in the experiments.

To decrease variability of the obtained results, all machines were configured so that their processors would work at their maximum frequencies and would not perform DVFS. Power measurements were obtained with a P4460 Kill A Watt EZ power meter, which has an accuracy (as reported by the manufacturer) of 0.5% and a refresh rate of one second. Experiments with the CubieTrucks consider both boards executing the same operations at the same time.

All tests were repeated five times in both architectures. The Kolmogorov-Smirnov [16] test was applied to each experiment results to test if time and power samples followed a normal distribution. This analysis has shown that the obtained results **do not** present a normal distribution. Hence we take the medians (and not the arithmetic means) and do not present error bars, as their computation assumes a normal distribution. The Dunn test [7] was used instead to

compare medians. Therefore, in the rest of this paper we only say there is a difference between results for two configurations if said difference was confirmed by the Dunn test.

3.1 Synthetic experiments

An MPI code was used to emulate a PFS data server activity. This emulation consists in receiving requests and performing the adequate operations to the local storage device. Due to the fact the CubieTruck can only run a modified Linux kernel, it was not possible to install a typical parallel file system. Moreover, we wanted to isolate the effects of the network and of the processing nodes from this analysis. This choice was made so that we could focus on the power and performance of each data server individually, ignoring the rest of the system.

Requests are provided to the server emulator through a file. The server reads the whole file first – so reading it will not interfere with the experiment – and then starts the execution. Each line of the file, representing a request, contains the request size and offset, in addition to a time difference relative to the previous request.

Synthetic workload files were created to emulate the data server point of view of a single client process accessing the parallel file system following different access patterns. Multiple instances of the server emulator were executed at the same time with different input files to have the effect of multiple clients concurrently accessing the same server. The tested access patterns, with two concurrent processes, include read and write tests, contiguous or 1D strided spatiality, and small (32KB) or large (4MB) requests. Each test accesses a 6GB file (3GB per process) from each server, so using a larger request size means generating fewer requests.

3.2 Hou10ni seismic application experiments

To validate our conclusions, we have also conducted experiments with the Hou10ni application. Hou10ni is a numerical code for the simulation of wave propagation in 3D heterogeneous acoustic and elastodynamic media, using Discontinuous Galerkin elements. In this work, we focused on acoustic media. It simulates the propagation of waves emitted by point sources and produces seismograms (the solution during the time at a given point) and snapshots (the solution in the whole domain at a given time). The computational domain is discretized by using tetrahedral meshes. The physical parameters are given on a structured grid and projected on the mesh.

In this experiment, we have considered a parallelepipedic domain $2150m \times 1440m \times 1220m$, discretized by a mesh of 898247 tetrahedra, using P1 elements. The density was assumed constant while the velocity of P waves varied in the computational domain from $1800m/s$ to $5000m/s$. The simulation time is $30ms$. We use 9 point sources placed near the surface of the geophysical domain and we wrote 4 snapshots (each $10ms$).

Hou10ni was executed on the Edel cluster from Grid’5000¹. All of the application’s I/O operations were made to the OrangeFS [6] parallel file system - version 2.8.7 - with two

¹<http://www.grid5000.fr/>

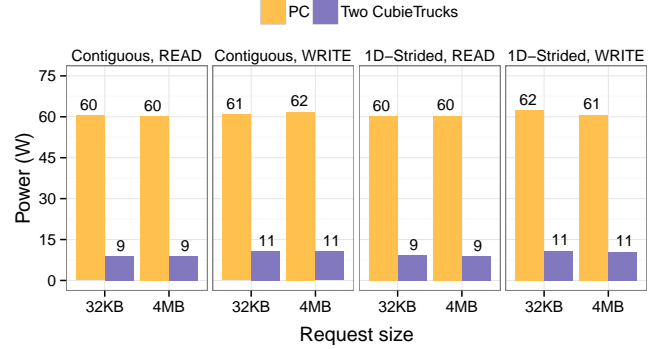


Figure 1: Mean power demand during benchmark executions

data servers and the default stripe size of 64KB. We have modified the data servers source code to include needed information (requests offset and size) to their trace files and collected traces from both servers in five different executions of the application. These traces were then provided to the emulator code in the CubieTrucks and in the PC, so the used machines would replicate the behavior of the real servers during those executions.

4. RESULTS

This section discusses results obtained by adhering to the methodology detailed in the previous section. First we analyze results for the synthetic workloads. Then, in Section 4.2 we present results for the Hou10ni seismic application.

4.1 Results with synthetic workloads

Figure 1 shows the mean power demand observed during the execution of the synthetic workloads. These values are obtained to each test by the arithmetic mean of instantaneous power measurements. The figure is separated by access spatiality (contiguous or 1D-strided) and operation (read or write). Each plot contains results for two request sizes (32KB or 4MB) and for both equipments: results with the **two CubieTrucks** in purple and with the PC in orange.

We can notice that **request size and access spatiality have no significant impact on power demand**. Nonetheless, **write tests have higher power demand than read ones**. This difference was confirmed by the Dunn test to all experiments except the 32KB contiguous and the 4MB strided tests in the PC. Power demand for writes is 23% (approximately 2W) higher in the CubieTrucks and 7% (approximately 4W) in the PC.

Compared to the PC, **the two CubieTrucks have power demand 82% lower for write experiments. The difference for reads is of 85%.**

4.1.1 Performance analysis

Figure 2 shows the bandwidth (in MB/s) achieved by the benchmarks. In this case, **request size and access spatiality had no significant impact on performance**. Bandwidth values are similar between read and write tests in the PC. On the other hand, **read bandwidth is 240%**

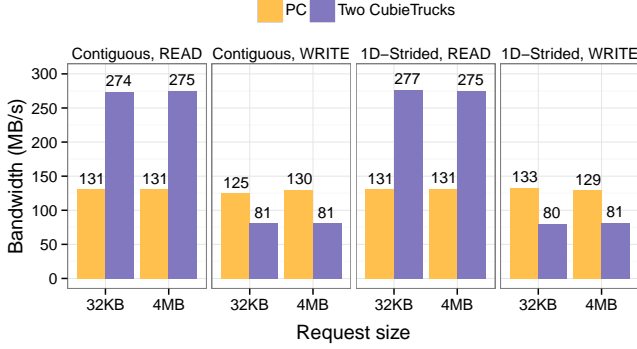


Figure 2: Performance observed in the benchmark executions

higher than write bandwidth in the CubieTrucks.

It is important to notice that two CubieTrucks are compared to a single PC, but in both the workload is the same, i.e., each CubieTruck writes or reads half the total amount of data (6GB). Therefore, **two CubieTrucks achieve roughly twice the read bandwidth achieved by the PC**, because each CubieTruck has read performance similar to the PC. **Write bandwidth is 37% lower in the CubieTrucks than in the PC.**

4.1.2 Energy efficiency

We use the **bytes per Joule** metric to quantify energy efficiency. It is the adequate metric for this discussion because it accounts for both energy consumption and performance.

Figure 3 shows the energy efficiency in MB/J on each architecture, separated by size and type of operation (write or read). It is again seen that the request size has little to no influence in the results. By the graphics, it is clearly observed that **the CubieTrucks offer higher energy efficiency**. The use of two low-power boards **increases efficiency by 1437% for read tests and by 287% for writes**.

4.1.3 Discussion

The discussion above has compared a low-power alternative for data servers - composed of two CubieTrucks - with a

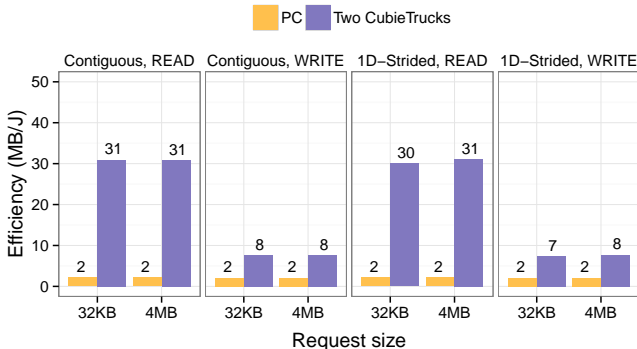


Figure 3: Energy efficiency observed in the benchmark executions

regular PC using an SSD for storage. Results have shown similar read performance in both architectures, with higher energy efficiency by the CubieTrucks due to a decrease 85% in power demand. Therefore, our results indicate that **replacing one regular data server by two CubieTrucks would double the read bandwidth while decreasing energy consumption by 85%.**

This alternative would increase energy efficiency for write workloads by 287% at the cost of a decrease in performance of 37%. Extrapolating the results obtained with two CubieTrucks points that **the regular server could be replaced by three or four CubieTrucks, decreasing energy consumption by 73% or 64%, respectively**. The first alternative assumes a performance decrease of 6%, while the second would increase write bandwidth by 25%. These replacements would further increase read performance.

As of September 2016, the equipments used in this research would cost approximately \$330 each SSD, \$100 each CubieTruck, and \$500 a similar PC. Therefore, the regular server would cost \$830, while the two low-power boards alternative would cost \$860. From the price perspective, the replacement would be viable and would further save money in electric bills. Moreover, the alternative would aggregate the storage capacity of more SSDs to the system.

The rest of this section validates these conclusions using a real application from the seismic field.

4.2 Results with Hou10ni

As discussed in Section 3, the Hou10ni seismic wave propagation simulation was executed in a cluster performing I/O to a parallel file system. Two data servers were used, and traces were collected during five application executions.

Each simulation took 27.05 minutes (median because results do not follow a normal distribution). From these, application spends median 11 minutes reading input data, and then the rest of its execution (a little over 16 minutes) writing results in periodic output phases.

Due to limitations of the power measurement equipment and to facilitate the experiments, from each trace we have separated the first minute - a read test - and the last minute - a write test - of the application execution. We have confirmed through non-parametric ANOVA that these windows are representative of the rest of the execution, regarding number of generated requests and amount of accessed data. Therefore, we can extrapolate results obtained with these portions to the whole application.

In these experiments, the two CubieTrucks emulate the two parallel file system data servers used during the experiments, while the single PC executes with traces from the first server only. Figure 4 shows the obtained results. Mean power demand, calculated from the arithmetic mean of instantaneous power measurements, is shown in Figure 4(a). Figure 4(b) presents the bandwidth, and finally Figure 4(c) presents energy efficiency in Megabytes per Joule. Each graph has results for the read and write portions of the application, and to each portion there are results with the two CubieTrucks in green and with the PC in red.

We can see the two CubieTrucks demand 86% less power

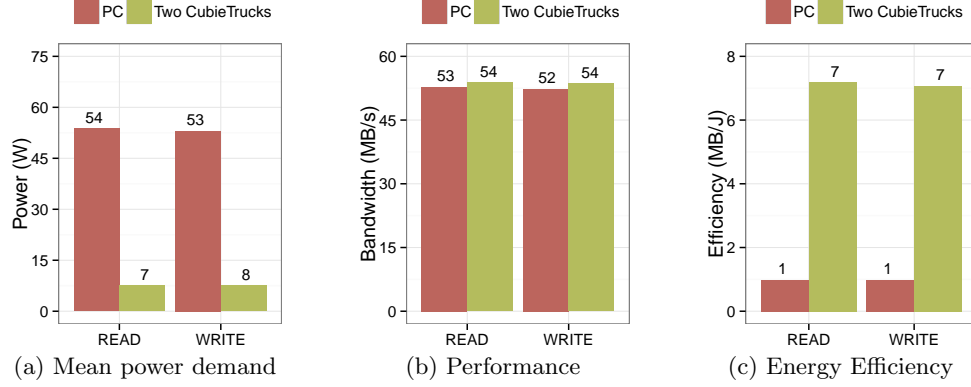


Figure 4: Results obtained with the Hou10ni application

than the PC, similarly to what was seen in Section 4.1. Power in this case is lower than what was observed with the benchmarks – 17% in the PC and 30% in the CubieTrucks. This difference comes from a less intensive workload, what can be confirmed if we compare bandwidth from Figure 4(b) with the previous performance results from Figure 2.

Both alternatives, **using the PC or the two CubieTrucks, resulted in the same performance.** This conflicts with previous results where the CubieTrucks provided lower write bandwidth. However, that did not happen here since the workload provided by the application is not as intensive as the synthetic benchmarks. Finally, **the two CubieTrucks provided seven times more energy efficiency.** This result can be directly extended to the application, which confirms the findings from Section 4.1 and highlights the potential of using low-power architectures as data servers for parallel file systems.

5. CONCLUSION AND FUTURE WORK

In this study, we aimed at evaluating the viability of replacing regular parallel file system servers by low-power architectures. The use of such low-power architectures for HPC has been the focus of some research (focusing on processing power), and they have also been receiving attention for adoption by cloud infrastructures and data centers. Nonetheless, we have not found any work in the literature where such alternative for storage servers is evaluated and its impact on performance quantified. Therefore, our main contribution is presenting an I/O performance and energy efficiency analysis of low-power servers compared to conventional architectures, considering representative access patterns.

Our results indicate that replacing one regular data server by two CubieTrucks would double the bandwidth and decrease energy consumption by 85% for read-intensive workloads. More low-power servers could replace a regular one to provide good trade-offs for write workloads as well. Furthermore, considering the system cost, it would be possible to acquire up to five ARM devices for the same price as a PC, while building a more energy-efficient platform.

To support our findings, in addition to the synthetic benchmark we have also investigated a seismic wave propagation application. In this scenario, we have observed similar per-

formance and 7 times more energy efficiency in the ARM-powered architecture when compared to the PC.

As future work, we plan to evaluate other scientific applications, with distinct access patterns. Moreover, the continuation of this research is the evaluation of scenarios including the network and more concurrent clients.

6. ACKNOWLEDGMENTS

This research received funding from the European EU H2020 Programme and from MCTI/RNP-Brasil in the context of the HPC4E Project, number 689772. Research was accomplished in the context of the LICIA joint laboratory. It was also partially funded by CNPq, CAPES, and FAPERGS.

The authors would like to thank Aishameriane Schmidt, from the Federal University of Santa Catarina, for her valuable help with statistical analysis.

7. REFERENCES

- [1] I. Alan, E. Arslan, and T. Kosar. Energy-aware data transfer algorithms. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis on - SC '15*, pages 1–12. IEEE, 2015.
- [2] H. Amur, J. Cipar, and V. Gupta. Robust and Flexible Power-Proportional Storage. *Proceedings of the 1st ACM symposium on Cloud computing*, pages 217–228, 2010.
- [3] E. V. Carrera, E. Pinheiro, and R. Bianchini. Conserving disk energy in network servers. In *Proceedings of the 17th annual international conference on Supercomputing - ICS '03*, pages 86–97. IEEE, 2003.
- [4] R. R. Chandrasekar, A. Venkatesh, K. Hamidouche, and D. K. Panda. Power-check: An energy-efficient checkpointing framework for HPC clusters. In *Proceedings - IEEE/ACM 15th International Symposium on Cluster, Cloud, and Grid Computing, CCGrid 2015*, pages 261–270. IEEE, 2015.
- [5] Cubieboard. <http://cubieboard.org/model/>. Accessed: Oct 2016.

- [6] Dell. Orangefs reference architecture. Technical report, 2012.
- [7] O. J. Dunn. Multiple comparisons among means. *Journal of the American Statistical Association*, 56(293):52–64, 1961.
- [8] R. Ge. Evaluating parallel I/O energy efficiency. In *Proceedings - IEEE/ACM Int. Conf. on Green Computing and Communications, GreenCom 2010 & Int. Conf. on Cyber, Physical and Social Computing, CPSCoM 2010*, pages 213–220. IEEE, 2010.
- [9] R. Ge, X. Feng, and X. H. Sun. SERA-IO: Integrating energy consciousness into parallel I/O middleware. In *Proceedings - 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, CCGrid 2012*, pages 204–211. IEEE, 2012.
- [10] S. Gurumurthi, A. Sivasubramaniam, M. Kandemir, and H. Franke. DRPM: dynamic speed control for power management in server class disks. In *30th Annual International Symposium on Computer Architecture, 2003. Proceedings.* ACM, 2003.
- [11] S. Hosseinimotlagh, F. Khunjush, and S. Hosseinimotlagh. A cooperative two-tier energy-aware scheduling for real-time tasks in computing clouds. In *2014 22nd Euromicro Int. Conf. on Parallel, Distributed, and Network-Based Processing*, pages 178–182, Feb 2014.
- [12] S. Ibrahim, T.-D. Phan, A. Carpen-Amarié, H.-E. Chihoub, D. Moise, and G. Antoniu. Governing energy consumption in hadoop through cpu frequency scaling: An analysis. *Future Generation Computer Systems*, 54:219–232, 2016.
- [13] J. Kim, J. Chou, and D. Rotem. Energy proportionality and performance in data parallel computing clusters. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 6809 LNCS:414–431, 2011.
- [14] P. Kogge, K. Bergman, S. Borkar, D. Campbell, W. Carson, W. Dally, M. Denneau, P. Franzon, W. Harrod, and K. Hill. Exascale Computing Study: Technology Challenges in achieving Exascale Systems. Technical report, Defense Advanced Research Projects Agency (DARPA IPTO), 2008.
- [15] W. Y. Lee. Energy-saving dvfs scheduling of multiple periodic real-time tasks on multi-core processors. In *Proceedings - 13th IEEE/ACM International Symposium on Distributed Simulation and Real Time Applications, DS-RT '09*, pages 216–223. IEEE Computer Society, 2009.
- [16] H. W. Lilliefors. On the kolmogorov-smirnov test for normality with mean and variance unknown. *Journal of the American Statistical Association*, 62(318):399–402, 1967.
- [17] J. Mair, Z. Huang, D. Eysers, and Y. Chen. Quantifying the energy efficiency challenges of achieving exascale computing. In *Proceedings - IEEE/ACM 15th Int. Symp. on Cluster, Cloud, and Grid Computing, CCGrid 2015*, pages 943–950. IEEE, 2015.
- [18] M. Nijim, A. Manzanarez, X. Ruan, and X. Qin. Hybud: An energy-efficient architecture for hybrid parallel disk systems. In *Proceedings - International Conference on Computer Communications and Networks, ICCCN*, pages 1–6. IEEE, 2009.
- [19] A.-C. Orgerie, M. Dias de Assuncao, and L. Lefevre. A Survey on Techniques for Improving the Energy Efficiency of Large Scale Distributed Systems. *ACM Computing Surveys*, pages 1–35, 2013.
- [20] E. L. Padoin, L. L. Pilla, M. Castro, F. Z. Boito, P. O. A. Navaux, and J.-F. Mehaut. Performance/energy trade-off in scientific computing: The case of ARM big.LITTLE and Intel Sandy Bridge. *IET Computers & Digital Techniques*, 2(3):1–14, 2014.
- [21] J. Peraza, A. Tiwari, M. Laurenzano, L. Carrington, and A. Snaveley. PMAc’s green queue: a framework for selecting energy optimal DVFS configurations in large scale MPI applications. *Concurrency and Computation: Practice and Experience*, pages 1–20, 2013.
- [22] N. Rajovic, A. Rico, N. Puzovic, C. Adeniyi-Jones, and A. Ramirez. Tibidabo1: Making the case for an ARM-based HPC system. *Future Generation Computer Systems*, 36:322–334, 2014.
- [23] Ambedded Technology Co. Software defined storage powered by arm based microserver cluster. <http://www.ambedded.com.tw/solutions.php>. Accessed: Oct 2016.
- [24] T. Saito, K. Sato, H. Sato, and S. Matsuoka. Energy-aware I/O optimization for checkpoint and restart on a NAND flash memory system. In *Proceedings - 3rd Workshop on Fault-tolerance for HPC at extreme scale - FTXS '13*, pages 41–48. ACM, 2013.
- [25] P. Shang and J. Wang. A novel power management for CMP systems in data-intensive environment. In *Proceedings - 25th IEEE International Parallel and Distributed Processing Symposium, IPDPS 2011*, pages 92–103. IEEE, 2011.
- [26] Y. Sverdlik. Paypal deploys arm servers in data centers, 2015. Accessed: Oct 2016.
- [27] B. Welch and G. Noer. Optimizing a hybrid SSD/HDD HPC storage system based on file size distributions. In *2013 IEEE 29th Symposium on Mass Storage Systems and Technologies (MSST)*, pages 1–12, 2013.
- [28] A. Younge, G. von Laszewski, L. Wang, S. Lopez-Alarcon, and W. Carithers. Efficient Resource Management for Cloud Computing Environments. In *IEEE International Green Computing Conference*, Chicago, IL, USA, 2010.
- [29] Q. Zhu, Z. Chen, L. Tan, Y. Zhou, K. Keeton, and J. Wilkes. Hibernator: Helping Disk Arrays Sleep Through the Winter. In *Proceedings - 20th ACM Symp. on Operating Systems Principles*, pages 177–190. ACM, Oct. 2005.